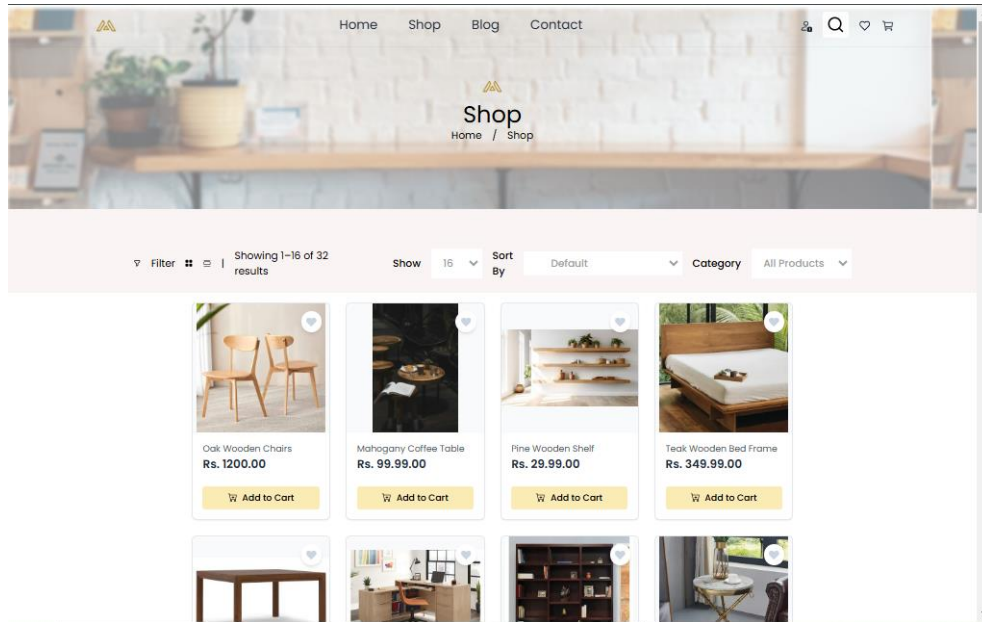


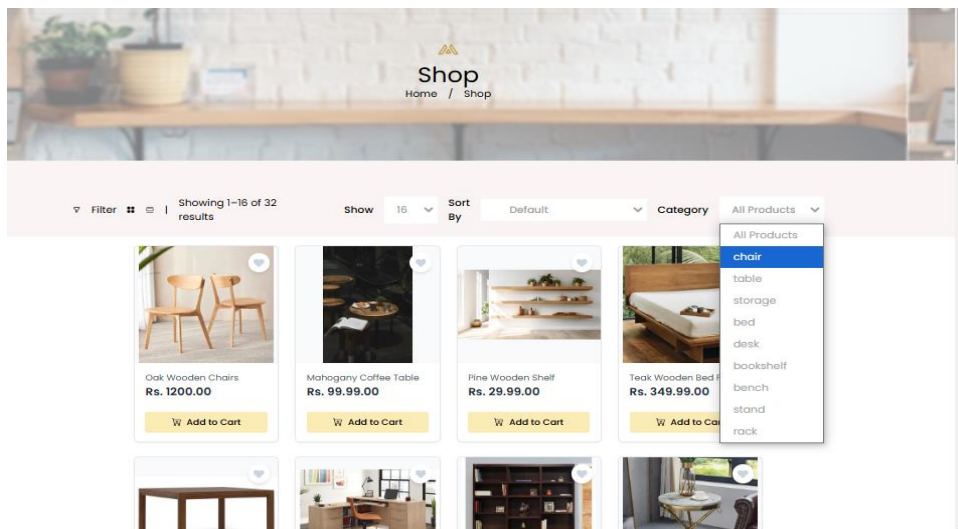
## Day 4 - Dynamic Frontend Components–Furniro

### ScreenShot of:

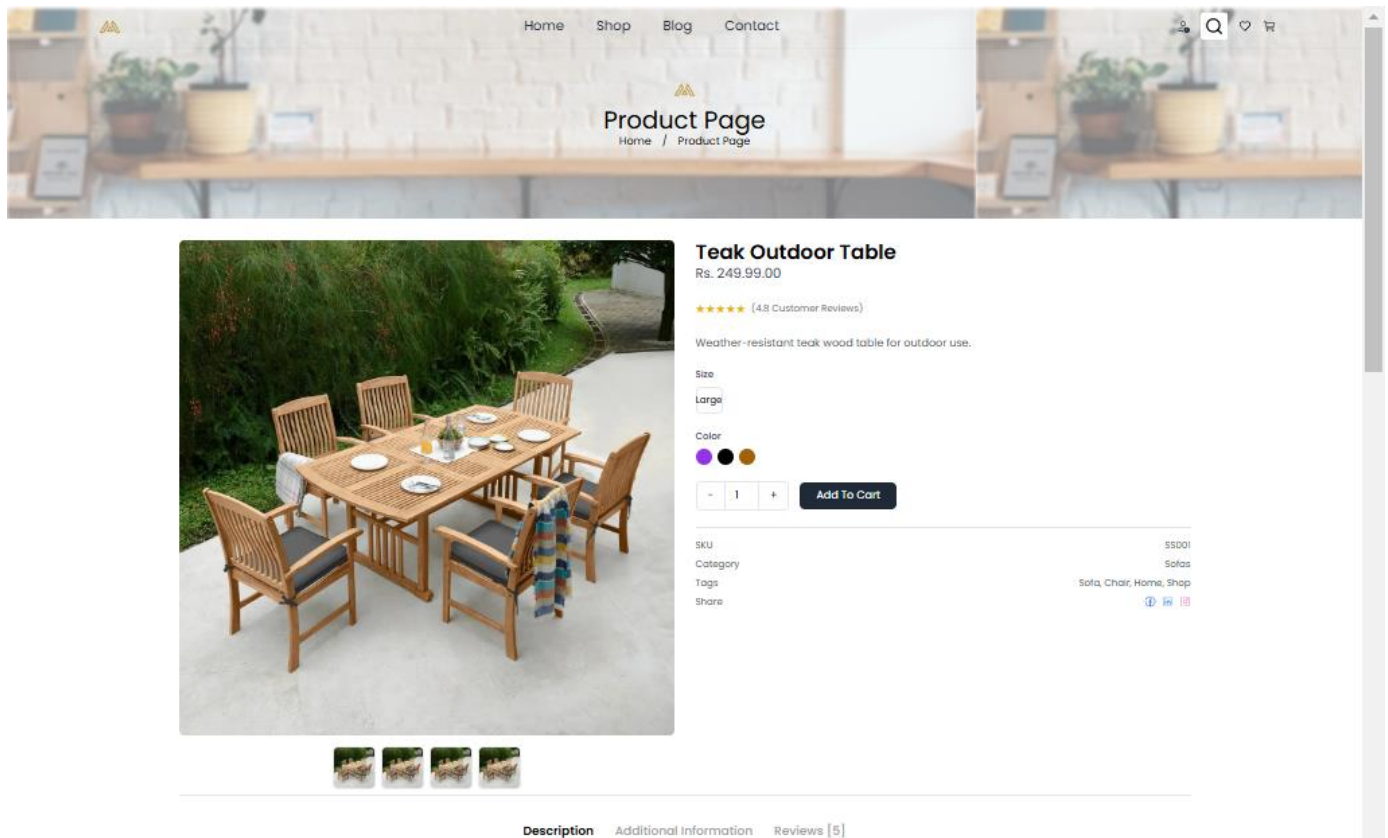
#### **\*Product Listing Component:**



#### **\* Category Component:**



## ***\*Product Detail Component:***

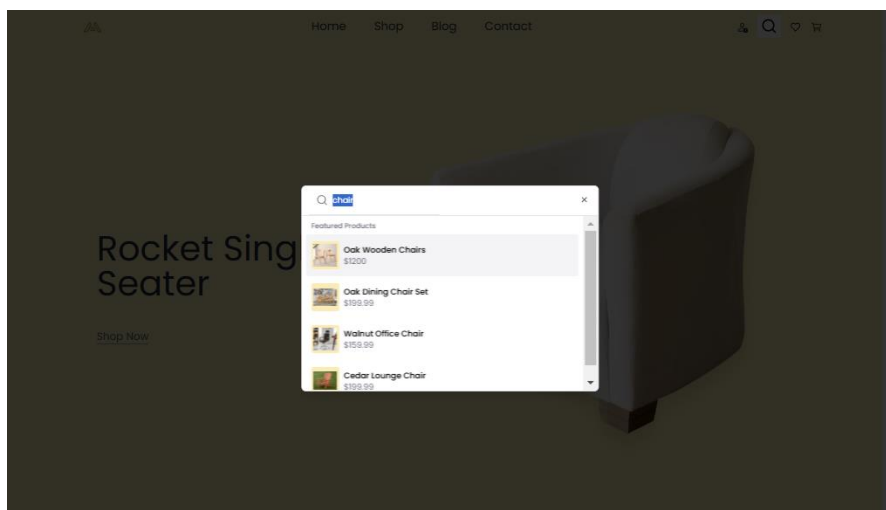


## ***\*Search Bar:***

***This Screenshot is of the Nav bar .***



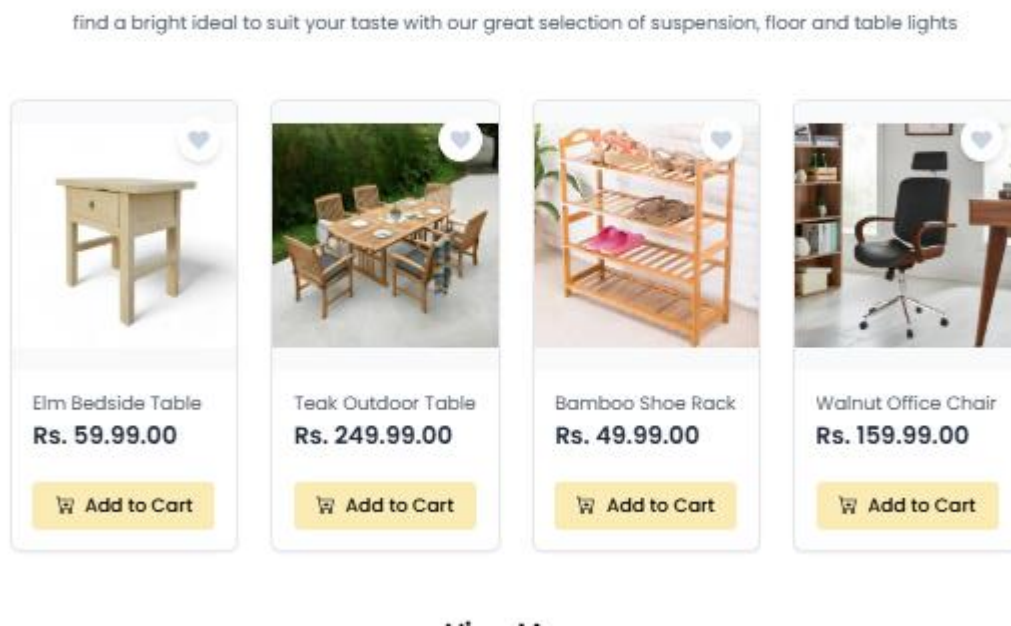
This is from Shop page where user can find products through Search Bar



### **\*Similar Products:**

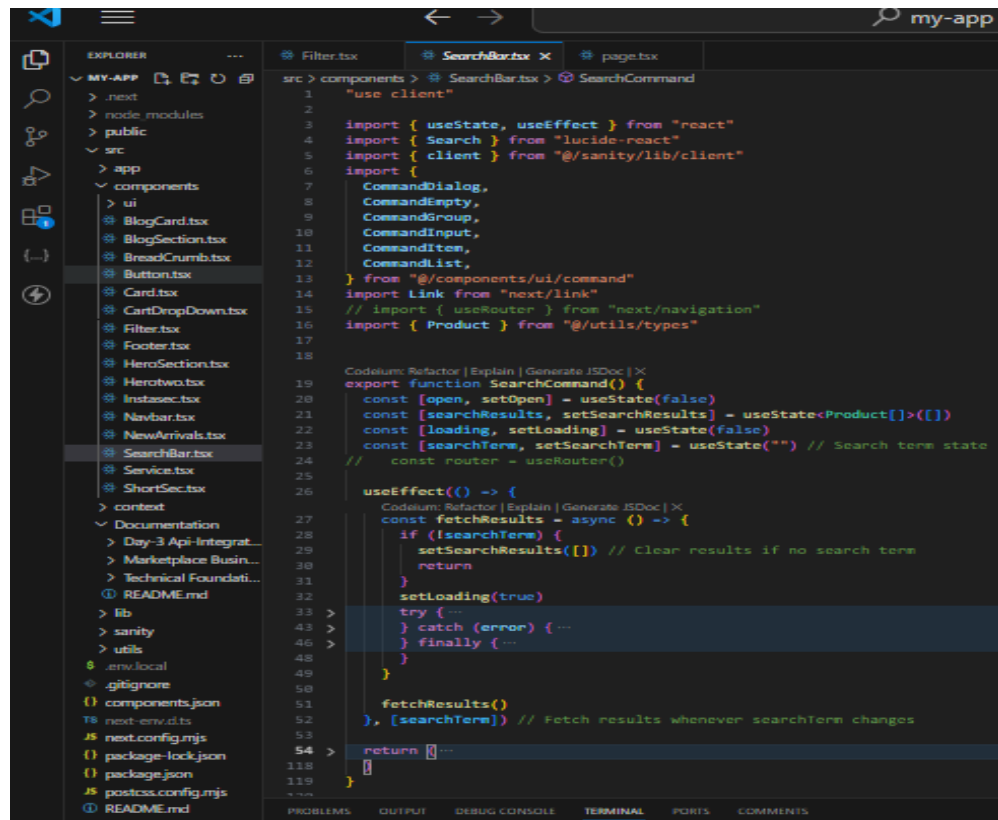
. Find this similar products by same tags through API.

## Top Picks for You



- A search bar is implemented using the `SearchCommand` component.
- The `handleSearch` function filters products based on the search query. It checks the `name` field of each product and show them.
- The search bar dynamically updates the `filteredData` as the user types.

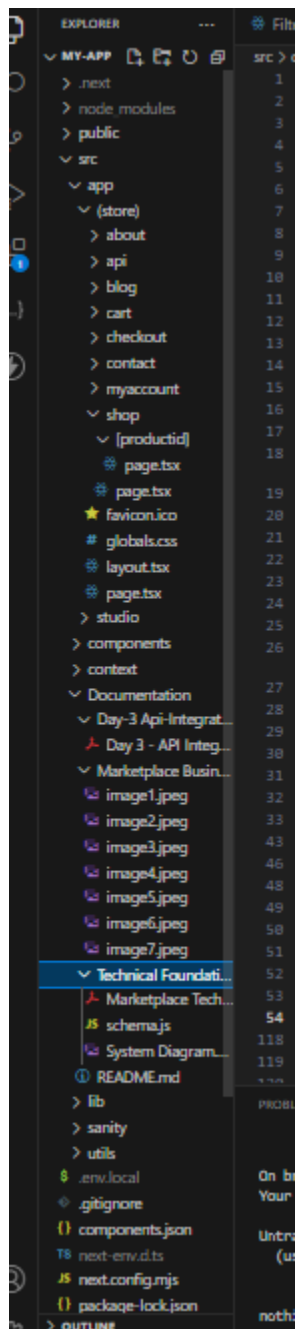
### ***Code:***



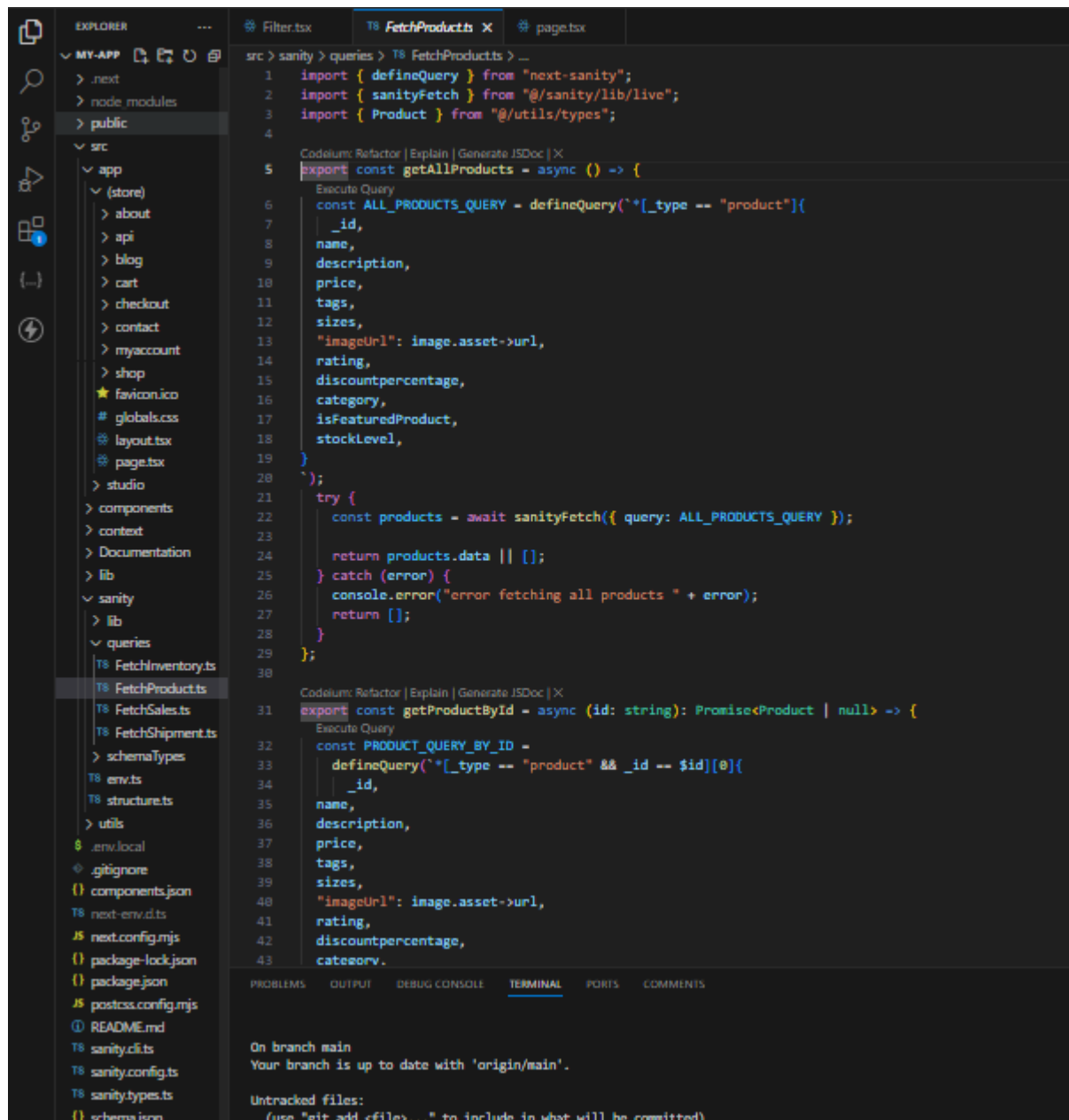
## **Product Details (Dynamic Routes):**

- Shows a product image, name, description, price and a discount percentage for real-time.
- Includes a "View Product" button linking to the product's detailed page using its `_id`.

## **File Structure:**



**Code:**



## **Product List:**

- Fetches data from a Sanity CMS backend for items of type "food".
- Displays each product in a card format, showing its name, description, price, discount, and an image.

## Dynamic Routing

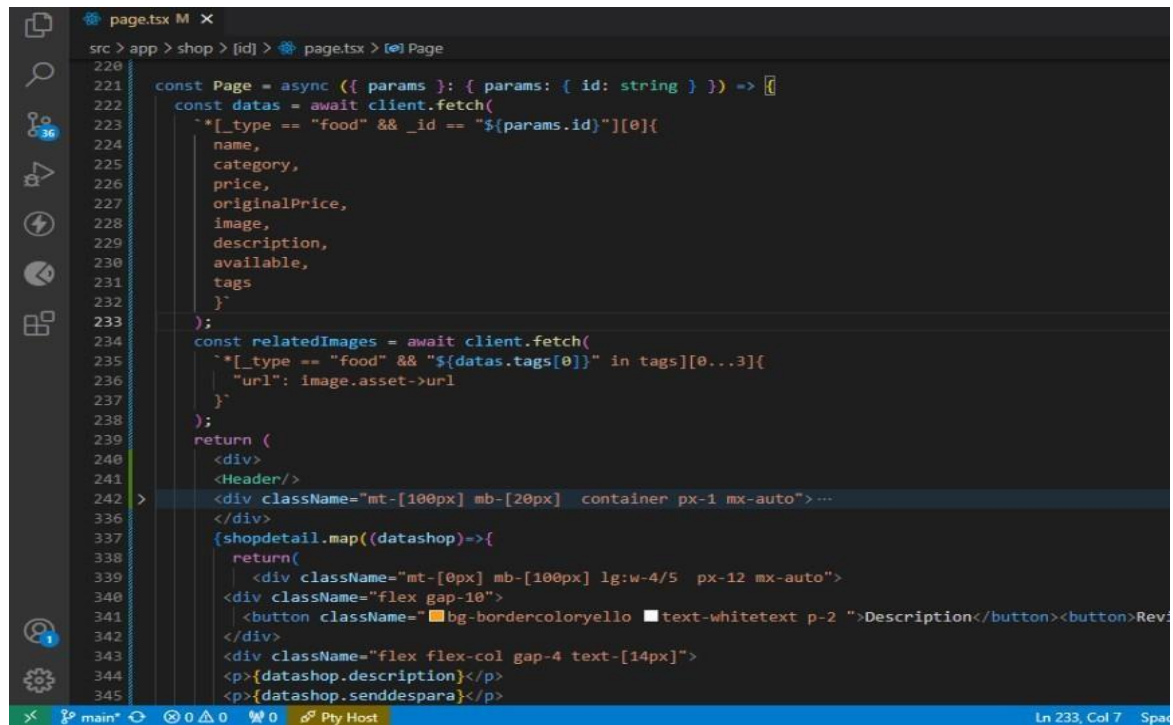
Dynamic routing is used to handle routes like `/shop/:id` for individual product pages.

### 1. Create Dynamic Route File:

In the `pages` directory, create a `[id].tsx` file inside the `shop` folder:

`pages/shop/[id].tsx`

### 2. Get Product Data Dynamically:



```
220
221 const Page = async ({ params }: { params: { id: string } }) => {
222   const datas = await client.fetch(
223     `*_type == "food" && _id == "${params.id}"`)[0]{
224     name,
225     category,
226     price,
227     originalPrice,
228     image,
229     description,
230     available,
231     tags
232   }
233 };
234 const relatedImages = await client.fetch(
235   `*_type == "food" && "${datas.tags[0]}" in tags`)[0...3]{
236   | "url": image.asset->url
237   }
238 };
239 return (
240   <div>
241     <Header/>
242     <div className="mt-[100px] mb-[20px] container px-1 mx-auto">...
243   </div>
244   <shopdetail.map((datashop)=>{
245     return(
246       <div className="mt-[0px] mb-[100px] lg:w-4/5 px-12 mx-auto">
247         <div className="flex gap-10">
248           <button className="bg-bordercoloryellow text-whitetext p-2">Description</button><button>Rev:
249         </div>
250         <div className="flex flex-col gap-4 text-[14px]">
251           <p>{datashop.description}</p>
252           <p>{datashop.senddespara}</p>
253         </div>
254       </div>
255     )
256   })
257 )
```

## Final CheckList

Frontend Component Development	Styling and Responsiveness	Code Quality	Documentation and Submission	Final Review
✓	✓	✓	✓	✓